

PST & PARR: Plan Specification Tools And
A Planning And Resource Reasoning Shell
For Use In Satellite Mission Planning

David McLean and Wen Yen

Software and Engineering Services
Bendix Field Engineering Corporation
Seabrook, Maryland

ABSTRACT

PST (Plan Specification Tools) is a set of tools which allows the user to specify satellite mission plans in terms of satellite activities, relevant orbital events and targets for observation. The output of these tools is a set of knowledge bases and environmental events which can then be used by PARR (Planning And Resource Reasoning shell) to build a schedule. PARR is a reactive planning shell which is capable of reasoning about actions in the satellite mission planning domain. This paper describes each of the PST tools and PARR and then briefly describes the use of PARR for scheduling computer usage in the multisatellite operations control center at Goddard Space Flight Center.

INTRODUCTION

PST has evolved through need in response to the problem of building planning systems which can be built quickly and which can respond to changing requirements after the initial system has been delivered. PST generates planning knowledge bases and resource viewing periods which can then be used by PARR. Experience with the ERBS-TDRSS (Earth Radiation Budget Satellite - Tracking and Data Relay Satellite System) Contact Planning System [McLean, 1987] has revealed the need for easy to use tools which allow operations personnel versus knowledge engineers to make changes to the planning knowledge bases when unforeseen (testing) situations arise. In the future, some satellite users (Extreme Ultra Violet Explorer) will make a preliminary survey and then make ad hoc selection of targets to be observed. PST is a step toward satisfying both of these requirements. In its present form, PST consists of an event specification tool, a tool to predict the satellite orbit positions, a target selection tool and a tool to generate viewing periods for resources and targets.

PARR is a single tool which can be invoked in a batch or interactive mode to generate a schedule for the events specified by PST. PARR is a reactive [Ow, 1988] planning shell because it has the ability to react intelligently to conflicts as they are encountered. This reactive component of PARR and the fact that the output of PARR is an actual conflict-free schedule makes it a tactical planning tool. On the other hand, PST can be thought of

as a set of strategic tools because PARR is used to execute the "strategic plans" (planning knowledge bases and environment) generated by the PST tools. These tools are integrated by a Menu-based EXecutive (MEX) which is based on the notion of menu-based natural language understanding introduced by Tennant [1983]. Figure 1 shows the organization of these tools.

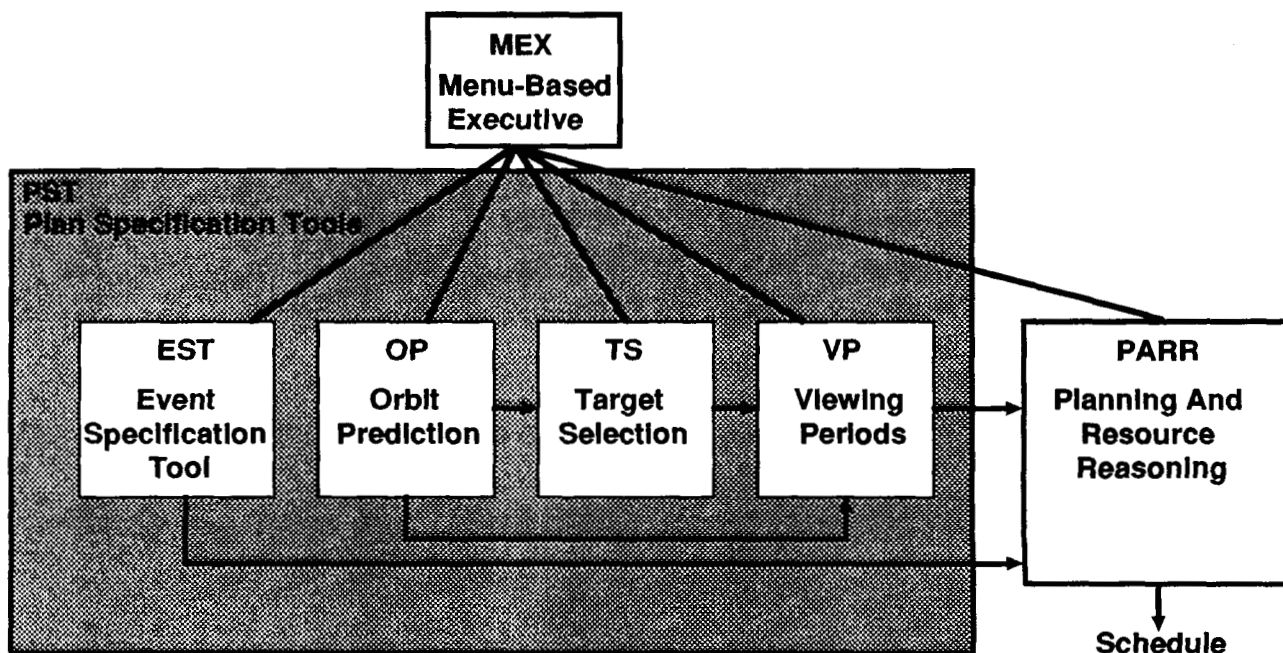


Figure 1 Planning Tools Environment

All of these tools are written in C and therefore are easily portable to a large variety of conventional hardware. Each of these tools will now be described and then an example using PARR for scheduling satellite equipment will be given.

THE EVENT SPECIFICATION TOOL

Both the MEX and the Event Specification Tool (EST) use a library of menu tools which allow the system designer to specify a grammar of valid commands. When using the menu tools, the designer specifies a top level menu which consists of one or more options (lines of text) each of which consists of a number of tokens (words delimited by spaces). Each of these tokens may then be the name of another menu. As options are selected, tokens which match the name of other menus get expanded into further menus and those which do not get added to the command line being built. This simple paradigm, combined with a few basic menu types, results in a very powerful tool for command line building. Some of the nonstandard menu types which are

typically used by EST are a matrix type which allows the designer to arrange the menu options in a matrix (two dimensional array) and a get-typed-text type which accepts text typed from the keyboard. A complete description of how to use the library of menu types is described in the MEX user's guide [1988].

The top level menu used by EST specifies the general plan specification model to be used. Currently only one model is used to specify events and this model includes the following components:

1. The spacecraft event name.
2. How often the event is to be scheduled (every N orbits).
3. The duration of the event.
4. When to start the event.
(usually with respect to a specified resource window)
5. Alternative means of scheduling (if any).
6. Spacecraft resources (tape and power, if any).
7. Constraints (if any).
8. Miscellaneous parameters such as priority and offset.

The actual top level menu looks like this:

```
main
  s/c_event every_n duration start_event any_alternatives \
  any_resources any_constraints misc_parms
```

Note that there is exactly one token for each component of the model (the backslash character indicates line continuation). Each of these tokens is a menu name which gets expanded when EST is invoked, for example, the following menus get used when the "start_event" token is expanded:

```
start_event
  title: Select the start time type you wish to specify.

  at_orbit_event_point
  user_specified_start
.
at_orbit_event_point
  title: Select START or END of the orbit point.

  START orbit_event
  END orbit_event
.
orbit_event
  title: Select an orbital view period.

  Daylight
  SAA
  MA TDRSS_viewing_periods
  SSA TDRSS_viewing_periods
```

```
.
user_specified_start
  prompt: Enter a start time (HH:MM:SS):
.
```

The "title:" token is used to specify text which is to be used to label the menu. The "prompt:" token is used to specify text to prompt the user for typed input. The actual number of user-specific menus, such as "s/c_event" and "orbit_event", required for an application using this model is small because most of the menus are generic to the model without being user specific.

The knowledge base structure which results from the use of EST is an event frame with slots for each component of the model. These components have been described elsewhere [The IEPS Knowledge Base Author's Manual, 1986], but an example of such an event frame demonstrates the correspondence between it and the current EST model:

```
TDRS_W_Tape_Dump_via_MA_antenna
;   priority
;       3
;   repeat every N views
;       2
;   duration
;       0:30
;   offset from view start
;       0:00
;   allow shift (if subsequent conflict occurs)
;       YES
;   resources:
;       LCS_POWER 200 1000
;       CR_TAPE -1 1500
;
;   constraints:
;       inview eq TDRS_W_MA_viewing_periods
;       inview ne South_Atlantic_Anomaly
;       activity eq NONE
;
;   event start:
;       START TDRS_W_MA_viewing_periods
;   alternative strategies:
;       EVENT TDRS_W_Tape_Dump_via_MA_antenna
;       NEXT 1
;
;   substructure
;       NONE
;   display text: (text displayed to user on expansion)

This is a tape dump using the
TDRS west multiple access antenna.
```

More detailed explanation of some of these slots will be described in the section on PARR.

THE SATELLITE ORBIT PREDICTION TOOL

The Flight Dynamics Facility (FDF) at Goddard provides accurate positional information for each of the standard orbital events, such as satellite daylight and TDRS viewing periods. However, because future missions will have the requirement of predicting where and when ad hoc targets of opportunity will be in view, some means of predicting viewing periods for these events is necessary. The satellite orbit prediction tool is the first step toward solving this problem. This tool is used to create a file which contains the positions of a satellite in an inertial geocentric coordinate system. These positions are generated at one minute intervals for a period of up to one week. This information is then used by the target selection tool and the resource and target viewing period tool to generate the necessary viewing periods for a variety of orbital phenomena.

The input to the satellite orbit prediction tool is a recent set of Brouwer mean elements obtained from FDF. The satellite positions are then predicted using Brouwer's first order secular perturbations due to an aspherical earth. For low orbiting satellites, such as ERBS, an empirical term simulating the effects of atmospheric drag is added. The accuracy of these predictions (for ERBS) is within 30 seconds after a week's set of predictions. Greater accuracy can be achieved by obtaining satellite orbit predictions directly from FDF, however a great deal of scheduling can be done with the lower level of accuracy.

THE TARGET SELECTION TOOL

The intent of this tool is to show some general ideas and capability which could be refined and expanded for actual mission requirements. The target selection tool consists of a user catalogue of potential targets (currently only fixed star-like targets are supported) and graphics software to display and select the targets. When the tool is invoked, the user is presented with a screen displaying the entire sky and all the user's targets which are visible on a given date and time. Those targets which are occulted by the earth are not visible. In addition, the position of the sun and the current slew point (instrument viewing direction) is displayed. The user may then step forward in time until the desired time of observation is reached and then select a subset of targets for planned observation by creating a window which contains the desired targets.

Once the targets are selected they are connected by lines which represents an "optimum" slew path. This slew path is arrived at

by using a Hopfield net [1986] and various heuristics which result in an acceptable solution to the traveling salesperson problem. At this point, slew times (based on relative right ascension and declination) and exposure times (based on magnitude) are calculated for each selected target. The targets are then written to a file in the optimum slew order with their respective slew and exposure times. This output file of selected targets can then be processed by the resource and target viewing period tool to generate viewing periods.

THE RESOURCE AND TARGET VIEWING PERIOD TOOL

The resource and target viewing period tool uses the information generated by the satellite orbit prediction tool and the target selection tool and generates viewing periods for selected orbital events and targets. These viewing periods become the environment within which PARR must expand the strategic plans (event frames) produced by EST. The South Atlantic Anomaly (SAA) position and size, orbit number, and the mean elements for TDRS east and TDRS west are obtained from FDF at Goddard. The position of the sun is calculated from formulae given in the Astronomical Almanac [1989]. In addition, the right ascension and declination of the selected targets is available and the user specifies which orbital events are required for scheduling (such as TDRS, Daylight and SAA) by menu selection. From this information, and from the previously calculated satellite positions, the viewing periods for each of these orbital events can be calculated.

The resource types available with this tool include the orbital events listed above and additional spacecraft resources such as power and tape. In the current prototype, spacecraft resources are initially assumed to be completely unused and available throughout the entire duration of the planning interval. A more detailed description of how spacecraft resources are represented is described in the section describing PARR.

The output file created by this tool lists the name of each resource preceded by the key letter 'R' and each of the targets preceded by the letter 'T'. These key letters let PARR know what the general resource type is so that when it executes the strategic plan the appropriate actions will be applied. Additional information follows each resource name depending upon the type, for example, slew and exposure times for each target. On the lines after each event name are listed the date, start and stop times and any additional information such as the orbit number.

THE PLANNING AND RESOURCE REASONING SHELL

PARR is the product of an evolving planning shell which has been in use by ERBS since May of 1987. Its capabilities have

therefore evolved from the requirements set by its actual use as well as from the anticipated needs of future users. There are currently two versions of PARR, one which runs on an IBM-PC using character graphics and another which runs on a UNIX-based workstation using bit-mapped graphics via X-Windows. PARR has three main modes of operation: batch, merge and interactive. In the batch and merge modes, the system builds a conflict-free schedule from the strategic knowledge bases alone, without consulting the user. The merge mode differs from the batch mode in that it is used to merge independent previous schedules. In the interactive mode PARR behaves as an intelligent assistant which not only handles the tedious details of planning requests but also suggests expert means for resolving conflicts when they arise. A great deal of the recent capability of PARR is due to its ability to reason about resources. The emphasis of this section will be to describe these recent capabilities.

Reasoning about resources requires that the various resource types be identified according to their behavior and then that a response (action) be selected which is most likely to solve the problem in view of the behavior of the resource. Therefore, a resource taxonomy which defines the possible behavior types is useful. Consider some typical resource examples used in spacecraft scheduling: sunlight availability, power for activities, fuel for maneuvering, tape for recording data and targets to observe, such as stars. These resources can be classified as follows:

Resource Example	Resource Type
Sunlight	Unlimited Capacity Subscribable (UCS)
Power	Limited Capacity Subscribable (LCS)
Fuel	Consumable and Nonreplenishable (CN)
Tape	Consumable and Replenishable (CR)
Star	Target

A UCS resource assumes that its presence alone is all that is required for its use. In the early stages of planning, TDRS availability is also a UCS resource because there is no knowledge of other spacecrafts' use of the system. LCS resources can support multiple requests with varying usage rates in parallel up to a threshold level of usage. For example, a tape recorder may use power at the same time that an instrument's sensing device is using it. CN can be considered a special case of the CR type, the case where replenishment is never scheduled. The CR resource type which PARR supports assumes that any number of requests can be supported in parallel and that they all consume at the same rate (as in a multichannel tape recorder). The resource type Target is really a special case of UCS and because targets require special treatment, a separate class has been made. The scheduling of targets is complex and therefore the discussion of this resource type will be deferred until after the other types

have been described more fully. Additional hybrid resource types can be constructed from combinations of the above; for example, battery power and "coolness" (a thermal sink) are a combination of LCS and CR because they behave like LCS resources which must be replenished from time to time.

Because PARR allows the user to specify resource attributes such as maximum capacity and usage rate, it can use this information as implicit resource constraints. Thus because PARR supports the above resource types, the user need not bother specifying the constraints for each resource explicitly. Normally, PARR uses these implicit constraints (such as, is the maximum capacity exceeded) when activities are requested by a user and rejects requests when these constraints are violated. However, because PARR can reason about these resource types, it can also try actions implicitly, such as replenishment when a CR resource is too low. Further, PARR frees the user from keeping track of the usage rates for each resource but allows the user to see the actual usage levels in the current situation.

The following discussion refers to the slots specified by the event frame and therefore it may be useful for the reader to refer the EST frame example given in a previous section. A UCS resource type is an example of a "viewing period" and because of this, a more traditional approach has been taken to represent this typical planning construct. In the current version of PARR, UCS resources such as sunlight may also be triggering events which are used to specify when an activity is to start. Thus, the name of the specific UCS resource may also appear as part of the "event start" slot. UCS resources are considered external to the spacecraft and therefore, if a particular spacecraft event requires that a UCS resource be available during the entire activity it should also be specified explicitly in the "constraints" slot. Because the constraints slot uses a syntax which specifies attribute-relation-value triplets, it is also useful for specifying which UCS resources should not be in view, such as the South Atlantic Anomaly.

Spacecraft resources which are internal (or on a platform) are specified in the "resources" slot and include the LCS and CR resource types. When specifying these resource types the user must prefix the resource name with "LCS_" or "CR_" so that PARR knows which type is being specified. The first parameter that follows the resource name is the usage rate and the second parameter is the maximum capacity (both are in user-defined units, CR usage is in units/minute). A negative usage rate for a CR type indicates a replenishment activity. Any number of resource types can be specified in this slot but the order is important if the first part (first three characters) of the unprefixed resource names match. This allows the user to specify alternative resource types to be used if the initial types are unavailable. Thus resource names such as the following are typical:

LCS_Power_A
LCS_Power_B
CR_Tape_A
CR_Tape_B

In this example, PARR will implicitly try obtaining the alternative resource "LCS_Power_B" if "LCS_Power_A" is unavailable.

Although the LCS resource type is sophisticated because of its ability to handle overlapping time requests, the CR type has a number of properties which make its implementation even more complicated. First, the amount of resource (for example tape) available must be known before and after each usage and each replenishment. The complication starts when a user wishes to interactively insert new activities which require tape. When this request is made, there may be plenty of tape to support the new request but not also the subsequent requests which have been previously scheduled. This condition (not enough tape for subsequent events) will result in PARR trying the implicit action "replenish". If replenishment is accomplished and all other event constraints are passed, then the request is accepted, otherwise it is denied. Another complication occurs when the user wishes to delete a replenishment. PARR will allow a deletion only if all subsequently scheduled events can be accomplished without the replenishment.

PARR represents targets as having the attributes slew time and exposure time which require special actions. The attribute slew specifies a duration during which an activity called slew must be scheduled before the actual "observation" can occur. The attribute exposure specifies the length of time that the target must be observed. Exposure time may be accumulated by scheduling the observation for multiple orbital viewing periods. PARR also has the ability to schedule multiple targets in a series, including slews and automatic tape replenishment. To specify the "observe targets" action in PARR, the user needs to have specified the slew and replenish (if required) events in addition to the observe event whose primary action will be indicated by the keyword TARGETS in the "event start" slot.

In addition to the implicit actions used by PARR to build the schedule specified in each of the event frames (goals), the user may also explicitly specify a set of heuristics which specify alternative actions (strategies) which may accomplish each goal. This combination of implicit and explicit actions and constraints is what makes PARR so powerful.

MSOCC EQUIPMENT SCHEDULING USING PARR

Part of the task of the Multisatellite Operations Control Center (MSOCC) is to schedule equipment (computers, etc) to support real

time contacts for a number of different satellite missions. This effort also includes scheduling the maintenance and offline testing for this equipment. To accomplish this, the MSOCC schedulers must first schedule all the real time support required for each of the missions and then schedule the maintenance and offline testing around the real time events. A further complication is that equipment assignment is changed weekly in order to insure that the same equipment units are not used to support the same missions week after week. This equipment change is necessary so that the reliability among each of the units within a given equipment type is maintained. When a particular unit fails a mission must be shifted to another unit which is equally reliable.

The task of entering the real time requests for support is done on a daily basis and is now accomplished by using a menu-based data entry system (based on the MEX library). Once this is done for each of the missions, PARR is used to assign equipment to each request and then merge all the real time requests into a single conflict free schedule. The equipment assignment is done by creating a knowledge base that represents the pattern of equipment assignment required for a particular week. There are eight possible equipment assignment patterns and therefore eight different knowledge bases. PARR represents the equipment as LCS resource types so that it is possible to support more than one mission on a particular piece of equipment. The user of the system selects the particular weekly pattern required before PARR is invoked and when conflicts of resources occur, PARR uses the alternative resources according to a particular mission's resource list. Any unresolvable conflicts are displayed to the user who may then reenter alternative real time requests. After this automatic assignment is done by PARR, the user may then "fine tune" the schedule by interactively moving resources for any desired mission event.

When the MSOCC schedulers are satisfied with the real time event schedule, a report generator is then invoked which prints each of the realtime events on timelines by equipment type and unit number. This report allows the schedulers to plan the maintenance and offline testing activities as required. After these activities are planned, they can also be put into a database via the same data entry tool as were the real time events. The only difference in data entry is that a different set of menus is used to interact with the user. Once all of the maintenance and offline activities have been entered, PARR is then invoked again to merge these activities with the merged real time events. Again PARR lets the user know if there are any conflicts that it cannot resolve. At this point the user can again make refinements to the merged schedule by interactive use of PARR until the schedule is satisfactory and then regenerate the timeline by equipment reports for further inspection. When schedules have been developed for each day of the week, another report generator is invoked which displays each of the

maintenance and offline teams (by name) along with their respective schedules for all the different equipment types they support.

A great deal of the success of the MSOCC equipment scheduling system is due to the flexibility of the data entry and report generation tools. However, it is interesting to note that PARR is general enough to be used in the "equipment scheduling" as well as in the "satellite scheduling" domain.

CONCLUSIONS

The development of these strategic (PST) and tactical (PARR) planning tools has proceeded with a number of goals in mind. Foremost is that these tools should be available and be of use to a variety of different users which have planning and scheduling requirements and which have at their disposal conventional hardware such as IBM-PCs or UNIX-based workstations. In addition, that these tools will evolve in a way that the user interface technology will remain relatively independent of the AI and algorithmic technology so that each can be maintained independently. This allows the greatest flexibility for taking advantage of new technology as it arrives. Finally, that the capabilities demonstrated by these tools should be derived from current mission support as well as the requirements for future support.

Most of the tools described above are not just prototypes but are in actual use. The ERBS-TDRSS Contact Planning System has recently been upgraded to support TDRS west requests as well as TDRS east and to use MEX and PARR. This has led to an enhanced ability to support ERBS because the complexity of specifying how constraint checking and resolution is to be accomplished has been greatly simplified and made more implicit. The fact that these tools are evolving through actual mission support as well as by prototyping features for future support gives credibility to this approach.

ACKNOWLEDGEMENT

The authors wish to thank Patricia Lightfoot and Carolyn Dent at NASA-GSFC/Code 514 and Ellen Stolarik at Bendix Field Engineering Corporation for their continual support of this work. This work was supported by NASA contracts NAS5-31000 and NAS5-27772.

REFERENCES

The Astronomical Almanac for the year 1989, U. S. Government Printing Office, Washington, D. C., 1989.

IEPS Knowledge Base Author's Manual. NASA-GSFC/Code 514, June 1988.

Hopfield, J. J., and Tank, D. W. "Computing with Neural Circuits: A model". Science, vol. 233, August 1986.

McLean, D. R., Littlefield, R. G., and Beyer D. S. "An Expert System for Scheduling Requests for Communications Links Between TDRS and ERBS". Telematics and Informatics, Vol. 4, No. 4, 1986.

MEX: A portable Menu-Based Executive. NASA-GSFC/Code 514, November 1988.

Ow P. S., Smith, S. F., and Thiriez, A. "Reactive Plan Revision". Proceedings of the AAAI-88 Seventh National Conference on Artificial Intelligence, Vol. 1, August 1988.

Tenant, H. R., Ross, K. M., Saenz, R. M., Thompson, C. W., and Miller, J. R. "Menu-based Natural Language Understanding", Proceedings of the Association for Computational Linguistics, MIT, June 1983.